## IN THE SPECIFICATION

The drawings were objected to because element 709 is not shown in Figure 7. The specification has been amended to remove numeral 709. Please amend the following paragraph on page 19, line 23 - page 20, lines 7 in the manner noted below. The amendments are believed to introduce no new matter.

In typical implementations, the generator program 705 can identify the selections and generate a logic description with information for implementing the various modules. The generator program 705 can be a Perl script creating HDL files such as Verilog, Abel, VHDL, and AHDL files from the module information entered by a user. In one example, the generator program identifies a portion of a high-level language program to accelerate. The other code is left for execution on a processor core. According to various embodiments, the generator program 705 identifies pointers and provides ports for each pointer. One tool with generator program capabilities is System on a Programmable Chip (SOPC) Builder available from Altera Corporation of San Jose, CA. The generator program 705 also provides information to a synthesis tool 707 to allow HDL files to be automatically synthesized. In some examples, a logic description is provided directly by a designer. Hookups between various components selected by a user are also interconnected by a generator program. Some of the available synthesis tools are Leonardo Spectrum, available from Mentor Graphics Corporation of Wilsonville, Oregon and Synplify available from Synplicity Corporation of Sunnyvale, California. The HDL files may contain technology specific code readable only by a synthesis tool. The HDL files at this point may also be passed to a simulation tool ~~709~~.

Please also amend the following paragraph on page 18, lines 4-17 in the manner noted below. The amendments are believed to introduce no new matter.

Figure 7 is a flow process diagram showing one example of a technique for implementing a programmable chip. At 701, primary components such as processor cores, Ethernet components, DMA controllers, etc., are identified for implementation on the programmable device. At 703, secondary components including peripherals such as memory components are identified. At 705, an interconnection component is generated. In one example, a bus fabric along with bus arbitration logic is generated. In another example, a simultaneous multiple primary component interconnection fabric is generated. At 707, data, address, and control lines are arranged to connect the primary and secondary components through the interconnection module. In one example, address, data, control, wait request, clock, and chip select lines are generated to allow communication between primary and secondary components. ~~At 709, d~~Data valid lines are also provided. In some instances, data valid lines are referred to as included in the set of control lines. A data valid line allows a secondary component to indicate to a primary component that data is available.